

## FAST SWEEPING ALGORITHMS FOR A CLASS OF HAMILTON–JACOBI EQUATIONS\*

YEN-HSI RICHARD TSAI<sup>†</sup>, LI-TIEN CHENG<sup>‡</sup>, STANLEY OSHER<sup>§</sup>, AND HONG-KAI ZHAO<sup>¶</sup>

**Abstract.** We derive a Godunov-type numerical flux for the class of strictly convex, homogeneous Hamiltonians that includes  $H(p, q) = \sqrt{ap^2 + bq^2 - 2cpq}$ ,  $c^2 < ab$ . We combine our Godunov numerical fluxes with simple Gauss–Seidel-type iterations for solving the corresponding Hamilton–Jacobi (HJ) equations. The resulting algorithm is fast since it does not require a sorting strategy as found, e.g., in the fast marching method. In addition, it provides a way to compute solutions to a class of HJ equations for which the conventional fast marching method is not applicable. Our experiments indicate convergence after a few iterations, even in rather difficult cases.

**Key words.** Hamilton–Jacobi equations, fast marching, fast sweeping, upwind finite differencing, eikonal equations

**AMS subject classifications.** 35, 65

**PII.** S0036142901396533

**1. Introduction.** Hamilton–Jacobi (HJ) equations have a rich pool of applications, ranging from those of optimal control theory and geometrical optics, to essentially any problem that needs the (weighted) distance function [14]. Examples include crystal growth, ray tracing, etching, robotic motion planning, and computer vision. Solutions of these types of equations usually develop singularities in their derivatives, and thus the unique viscosity solution [6] is sought.

In this article, we focus on the class of time independent HJ equations with Dirichlet boundary condition

$$H(\mathbf{x}, \nabla u) = r(\mathbf{x}), \quad u|_{\Gamma} = 0;$$

$H(\mathbf{x}, \mathbf{p})$  are strictly convex nonnegative, and  $\lim_{\lambda \rightarrow 0} H(\mathbf{x}, \lambda \mathbf{p}) = 0$ . We explain our method using the following important model equation:

$$(1.1) \quad H(\phi_x, \phi_y) = \sqrt{a\phi_x^2 + b\phi_y^2 - 2c\phi_x\phi_y} = r,$$

where  $\phi : \mathbb{R}^2 \mapsto \mathbb{R}$  is continuous and  $a, b, c$ , and  $r$  can be either constants or scalar functions; in the latter case,  $H$  depends also on  $x$ , defined on  $\mathbb{R}^2$ , satisfying  $ab > c^2$ ,  $a, b, r > 0$ . With  $a = b = 1$  and  $c = 0$ , we have the standard eikonal equation for

\*Received by the editors October 15, 2001; accepted for publication (in revised form) October 17, 2002; published electronically May 6, 2003.

<http://www.siam.org/journals/sinum/41-2/39653.html>

<sup>†</sup>Department of Mathematics and Program in Applied and Computational Mathematics, Princeton University, Princeton, NJ 08544 ([ytsai@math.princeton.edu](mailto:ytsai@math.princeton.edu)). The research of this author was supported by ONR N00014-97-1-0027, DARPA/NSF VIP grant NSF DMS 9615854, and ARO DAAG 55-98-1-0323.

<sup>‡</sup>Department of Mathematics, University of California at San Diego, La Jolla, CA 92093-0112 ([lcheng@math.ucsd.edu](mailto:lcheng@math.ucsd.edu)). The research of this author was supported by NSF grants 0112413 and 0208449.

<sup>§</sup>Department of Mathematics, University of California at Los Angeles, Los Angeles, CA 90095 ([sjo@math.ucla.edu](mailto:sjo@math.ucla.edu)). The research of this author was supported by ONR N00014-97-1-0027, DARPA/NSF VIP grant NSF DMS 9615854, and ARO DAAAG 55-98-1-0323.

<sup>¶</sup>Department of Mathematics, University of California at Irvine, Irvine, CA 92697-3875 ([zhao@math.ucla.edu](mailto:zhao@math.ucla.edu)).

Report Documentation Page			Form Approved OMB No. 0704-0188	
<p>Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.</p>				
1. REPORT DATE <b>MAY 2003</b>	2. REPORT TYPE	3. DATES COVERED <b>00-00-2003 to 00-00-2003</b>		
4. TITLE AND SUBTITLE <b>Fast Sweeping Algorithms for a Class of Hamilton-Jacobi Equations</b>			5a. CONTRACT NUMBER	
			5b. GRANT NUMBER	
			5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S)			5d. PROJECT NUMBER	
			5e. TASK NUMBER	
			5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) <b>Princeton University ,Department of Mathematics and Program in Applied and Computational Mathematics,Princeton,NJ,08544</b>			8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)			10. SPONSOR/MONITOR'S ACRONYM(S)	
			11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT <b>Approved for public release; distribution unlimited</b>				
13. SUPPLEMENTARY NOTES				
14. ABSTRACT <b>We derive a Godunov-type numerical flux for the class of strictly convex, homogeneous Hamiltonians that includes <math>H(p, q) = ap^2 + bq^2 - 2cpq</math>, <math>c^2 &lt; ab</math>. We combine our Godunov numerical fluxes with simple Gauss-Seidel-type iterations for solving the corresponding Hamilton-Jacobi (HJ) equations. The resulting algorithm is fast since it does not require a sorting strategy as found, e.g., in the fast marching method. In addition, it provides a way to compute solutions to a class of HJ equations for which the conventional fast marching method is not applicable. Our experiments indicate convergence after a few iterations, even in rather difficult cases.</b>				
15. SUBJECT TERMS				
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT <b>Same as Report (SAR)</b>	18. NUMBER OF PAGES <b>22</b>
a. REPORT <b>unclassified</b>	b. ABSTRACT <b>unclassified</b>	c. THIS PAGE <b>unclassified</b>	19a. NAME OF RESPONSIBLE PERSON	

which many numerical methods have been developed. This equation has the essential features of HJ equations with convex Hamiltonians, so that we can easily explain our algorithm, and is general enough that fast marching is not applicable.

In the following subsections, we will review some of the solution methods for the eikonal equation since it forms the motivation of our work. We then present a fast Gauss–Seidel-type iteration method for (1.1) which utilizes a monotone upwind Godunov flux for the Hamiltonian. We show numerically that this algorithm can be applied directly to equations of the above type with variable coefficients.

**1.1. Solving eikonal equations.** In geometrical optics [10], the eikonal equation

$$(1.2) \quad \sqrt{\phi_x^2 + \phi_y^2} = r(x, y)$$

is derived from the leading term in an asymptotic expansion

$$e^{i\omega(\phi(x,y)-t)} \sum_{j=0}^{\infty} A_j(x, y, t) (i\omega)^{-j}$$

of the wave equation

$$w_{tt} - c^2(x, y)(w_{xx} + w_{yy}) = 0,$$

where  $r(x, y) = 1/|c(x, y)|$  is the function of slowness. The level sets of the solution  $\phi$  can thus be interpreted as the first arrival time of the wave front that is initially  $\Gamma$ . It can also be interpreted as the “distance” function to  $\Gamma$ .

We first restrict our attention for now to the case in which  $r = 1$ . Let  $\Gamma$  be a closed subset of  $\mathbb{R}^2$ . It can be shown easily that the distance function defined by

$$d(\mathbf{x}) = \text{dist } (\mathbf{x}, \Gamma) := \min_{p \in \Gamma} \|\mathbf{x} - p\|, \quad \mathbf{x} = (x, y) \in \mathbb{R}^2,$$

is the viscosity solution to (1.2) with the boundary condition

$$\phi(x, y) = 0 \quad \text{for } (x, y) \in \Gamma.$$

Rouy and Tourin [20] proved the convergence to the viscosity solution of an iterative method solving (1.2) with the Godunov Hamiltonian approximating  $\|\nabla\phi\|$ . The Godunov Hamiltonian function can be written in the following form:

$$(1.3) \quad H_G(p_-, p_+, q_-, q_+) = \sqrt{\max\{p_-^+, p_+^-\}^2 + \max\{q_-^+, q_+^-\}^2},$$

where  $p_{\pm} = D_{\pm}^x \phi_{i,j}$ ,  $q_{\pm} = D_{\pm}^y \phi_{i,j}$ ,  $D_{\pm}^x \phi_{i,j} = \pm(\phi_{i\pm 1,j} - \phi_{i,j})/h$  and accordingly for  $D_{\pm}^y \phi_{i,j}$ , and  $x^+ = \max(x, 0)$ ,  $x^- = -\min(x, 0)$ .

Osher [13] provided a link to time dependent eikonal equations by proving that the  $t$ -level set of  $\phi(x, y)$  is the zero level set of the viscosity solution of the evolution equation at time  $t$ ,

$$\psi_t = \|\nabla\psi\| = 0,$$

with appropriate initial conditions. In fact, the same is true for a very general class of HJ equations (see [13]). As a consequence, one can try to solve the time dependent

equation by the level set formulation [17] with high order approximations on the partial derivatives [9, 18]. Crandall and Lions proved that the discrete solution obtained with a consistent, monotone numerical Hamiltonian converges to the desired viscosity solution [5].

Tsitsiklis [25] combined heap sort with a variant of the classical Dijkstra algorithm to solve the steady state equation of the more general problem

$$\|\nabla\phi\| = r(\mathbf{x}).$$

This was later rederived in [23] and also reported in [8]. It has become known as the fast marching method, whose complexity is  $\mathcal{O}(N \log(N))$ , where  $N$  is the number of grid points. Osher and Helmsen [15] have extended the fast marching-type method to somewhat more general HJ equations. We will comment on this in a following section.

**1.2. Anisotropic eikonal equation.** We return to the Hamiltonian in question:  $H(p, q) = \sqrt{ap^2 + bq^2 - 2cpq}$ . Writing the quadratic form as

$$ap^2 + bq^2 - 2cpq = \begin{pmatrix} p & q \end{pmatrix} \begin{pmatrix} a & -c \\ -c & b \end{pmatrix} \begin{pmatrix} p \\ q \end{pmatrix},$$

it is easy to see that we can diagonalize the symmetric matrix in the middle of the equation for our previously noted choices of  $a, b, c$  and find a coordinate system  $\xi$ - $\eta$  such that, after rescaling, the Hamiltonian becomes

$$H(\tilde{p}, \tilde{q}) = \sqrt{\tilde{p}^2 + \tilde{q}^2}.$$

The eigensystem of the above matrix defines the anisotropy. Indeed, the authors in [15] proposed to solve the constant coefficient equation (1.1) by first transforming it to (1.2) in the  $\xi$ - $\eta$  coordinate system.

This anisotropy occurs in fields such as ray tracing in special media, e.g., crystals, in which there are “preferred” directions. Furthermore, we will see that it can be a result of considering the geodesic distance function on a manifold  $M$  that is defined as the graph of a smooth function  $f$ .

Let  $\phi$  be the distance function such that

$$\phi(x, y) = \min_{\gamma \subset M} \int_{\gamma} ds$$

and  $\gamma$  connects the point  $(x, y)$  with the set  $\Gamma \subset M$ . The minimizing curve is called the geodesic, and  $\phi$  the distance function to  $\Gamma$  on  $M$ . Moreover,  $\phi$  solves

$$(1.4) \quad \|P_{\nabla\psi}\nabla\phi\|^2 = 1, \quad \phi|_{\Gamma} = 0,$$

where  $\psi(x, y, z) = f(x, y) - z$ , and the projection operator [4]

$$P_{\nabla\psi} = I - \frac{\nabla\psi \otimes \nabla\psi}{\|\nabla\psi\|^2},$$

which projects a vector onto a plane whose normal is parallel to  $\nabla\psi$ . Using the fact that  $P_{\nabla\psi}$  is a projection operator, a simple calculation shows that

$$(1.5) \quad \|P_{\nabla\psi}\nabla\phi\|^2 = \left(1 - \frac{f_x^2}{f_x^2 + f_y^2 + 1}\right) \phi_x^2 + \left(1 - \frac{f_y^2}{f_x^2 + f_y^2 + 1}\right) \phi_y^2 - 2 \frac{f_x f_y}{f_x^2 + f_y^2 + 1} \phi_x \phi_y.$$

This is clearly of the form of Hamiltonians that we are interested in. We will apply our algorithm to compute the geodesic distance later in this paper.

There are other approaches that are designed to compute distances on manifolds. For example, [11] provided an algorithm to compute the geodesic distance on triangulated manifolds. Barth [2] uses the discontinuous Galerkin method to find distance on graphs of functions that are represented by spline functions. In [4], the authors embed the manifold as the zero level set of a Lipschitz continuous function and solve the corresponding time dependent eikonal equation (1.4) in the embedding space. As we have mentioned in the previous subsection, the zero level set of the time dependent eikonal equation at time  $t_1$  is the  $t_1$ -level set of the solution to the stationary eikonal equations (see [13]). In [12], the authors adopted the standard fast marching method to solve the isotropic eikonal equation in a thin band of thickness  $\epsilon$ , which encloses the manifold  $M$ , and proved that the restriction of the solution to  $M$  converges to the geodesic distance as  $\epsilon$  goes to 0. In [21, 22], the authors provide an ordered upwind method to solve a general class of static HJ equations. We will comment on their method in a later subsection.

**1.3. Osher's fast marching criteria.** Since the fast marching method is by now well known, we will not give much detail on its implementation in this paper. In general, this involves a sorting procedure and the solution of

$$(1.6) \quad H_G(D_-^x \phi_{i,j}, D_+^x \phi_{i,j}, D_-^y \phi_{i,j}, D_+^y \phi_{i,j}) = 1$$

for  $\phi_{ij}$  in terms of its four neighboring values. More precisely, the heap sort strategy of the fast marching method requires a monotone update sequence. The updated value of a grid node has to be greater than or equal to those of the grid nodes used to form the finite difference stencil. This amounts to the condition

$$pH_p + qH_q \geq 0,$$

which dictates that the solution be nondecreasing along the characteristics. However, if we use one-sided upwind finite difference approximations for partial derivatives of  $\phi$  on a Cartesian grid, it is equivalent to demanding that the partial derivatives of  $\phi$  (i.e.,  $p$  and  $q$ ) and their corresponding components of the characteristics directions (i.e.,  $dx/dt$  and  $dy/dt$ ) have the same sign. Since  $dx/dt = H_p$  and  $dy/dt = H_q$ , we have the stricter Osher's fast marching criterion

$$(1.7) \quad pH_p \geq 0, \quad qH_q \geq 0.$$

It does not matter whether the Hamiltonian is convex or not; as long as criterion (1.7) is satisfied, a simple fast marching algorithm can be applied. But if the criterion is not satisfied, fast marching cannot be applied to the problem on a Cartesian grid. Of course there are Hamiltonians that do not satisfy (1.7). In the class of Hamiltonians that we consider, as long as  $c \neq 0$ , it is likely that the values of  $p$  and  $q$  differ to the extent that the above criterion is no longer satisfied. In light of criterion (1.7), we have also tried to find directions  $\xi(x, y)$  and  $\eta(x, y)$  locally in which  $\tilde{p}H_{\tilde{p}} \geq 0$ ,  $\tilde{q}H_{\tilde{q}} \geq 0$ . However, if one insists on using Cartesian grids, the implementation of this approach might be a bit hairy. We are interested especially in tackling, over a Cartesian grid, problems where the solution is nondecreasing along characteristics but where Osher's fast marching criterion is not satisfied.

**1.4. The sweeping idea.** Danielsson [7] proposed an algorithm to compute Euclidean distance to a subset of grid points on a two dimensional grid by visiting each grid node in some predefined order. In [3], Boué and Dupuis suggest a similar “sweeping” approach to solve the steady state equation which, by experience, results in an  $\mathcal{O}(N)$  algorithm for the problem at hand. This “sweeping” approach has recently been used in [24] and [27] to compute the distance function to an arbitrary data set in computer vision. In [26], the author provides some theoretical evidence indicating that sweeping converges to an approximate Euclidean distance function, i.e., to an approximate viscosity solution of  $|\nabla\phi| = 1$  in  $2d$  predetermined iterations. We will talk about these iterations in a later section. Using this “sweeping” approach, the complexity of the algorithms drops from  $\mathcal{O}(N \log N)$  in fast marching to  $\mathcal{O}(N)$ , and the implementation of the algorithms becomes a bit easier than the fast marching method that requires heap sort.

This sweeping idea is best illustrated by solving the eikonal equation in  $[0, 1]$ :

$$|u_x| = 1, \quad u(0) = u(1) = 0.$$

Let  $u_i = u(x_i)$  denote the grid values associated with the uniform grid composed of the gridpoints  $0 = x_0 < x_1 < \dots < x_n = 1$ . We then solve the discretized nonlinear system

$$(1.8) \quad \sqrt{\max(\max(D_- u_i, 0)^2, \min(D_+ u_i, 0)^2)} = 1, \quad u_0 = u_n = 0,$$

by our sweeping approach. We initially set  $u_i^{(0)} = \infty$ ,  $i = 1, \dots, n - 1$ . In practice,  $\infty$  can be replaced by some number  $K$ , which is larger than  $\max_{x \in [0, 1]} u$ . Let us begin by sweeping from 0 to 1; i.e., we update  $u_i$  from  $i = 1$  increasing to  $i = n - 1$ . This is “equivalent” to following the characteristics emanating from  $x_0$ . Let  $u_i^{(1)}$  denote the grid values after this sweep. We then have

$$u_i^{(1)} = \begin{cases} \frac{i}{n} & \text{if } i = 1, \dots, n - 2, \\ \frac{1}{n} & \text{if } i = n - 1. \end{cases}$$

Notice that at  $i = n - 1$ , we actually use the upwind information from the neighboring right boundary point. Furthermore, notice that  $u_i^{(1)}$  already has the correct desired values for  $i \leq n/2$  since the sweep goes from left to right, the desired upwind direction for these  $i$ . In the second sweep, we update  $u_i$  from  $i = n - 1$  decreasing to 1, starting with  $u_i^{(1)}$ . During this sweep, we follow the characteristics emanating from  $x_n$ . The use of (1.8) is essential, since it determines what happens when two characteristics cross each other. It is then not hard to see that, after the second sweep,

$$u_i = \begin{cases} \frac{i}{n} & \text{if } i \leq \frac{n}{2}, \\ \frac{(n-i)}{n} & \text{otherwise.} \end{cases}$$

Notice that the correct values at  $i \leq n/2$  derived after the first sweep are unchanged, and new and correct values for  $i > n/2$  are created. In summary, this simple iterative algorithm can be described as follows: at the  $k$ th iteration, solve

$$\max \left( \max \left( \frac{u_i^{(k)} - u_{i-1}^{(k-1)}}{\Delta x}, 0 \right), \min \left( \frac{u_{i+1}^{(k-1)} - u_i^{(k)}}{\Delta x}, 0 \right) \right) = 1$$

for  $u_i^{(k)}$  for each  $i$  going from 1 to  $n - 1$  in the first iteration ( $k = 1$ ), and from  $n - 1$  to 1 for the second iteration ( $k = 2$ ). However, for more complicated equations and boundary conditions, it is not so easy to write down the equivalent explicit solution.

In this paper, we will extend this sweeping approach to a class of HJ equations that cannot be solved by the fast marching algorithm, by first deriving a Godunov Hamiltonian.

In [21, 22], the authors proposed a one-pass method that is based on a control-theoretic viewpoint. In principle, they solve the HJB equation

$$(1.9) \quad \max_{\mathbf{a}} \nabla u \cdot \mathbf{a} f(\mathbf{a}, \mathbf{x}) = 1,$$

where  $\mathbf{p} = (p, q)$  and the function  $f(\mathbf{a}, \mathbf{x})$  is the speed of motion. This formula is the second Legendre transform taken on the sphere; see, e.g., [16, 19].

The idea is still to follow the characteristics and update the grid value in a monotone sequence. In a notation similar to the two dimensional setting of [21, 22], we let  $u_o$  be the grid value we are updating. To update  $u_o$ , we have to look for two other grid values  $u_r$  and  $u_s$ , which are not necessarily the immediate grid neighbors of  $u_o$ . For example, if  $u_o$  is the grid value  $u_{i,j}$ , the immediate neighbors of  $u_o$  are then  $u_{i+1,j}, u_{i,j+1}, u_{i-1,j}$ , and  $u_{i,j-1}$ . As we indicated in the previous subsection, it is possible that  $u_o$  is less than all its immediate neighboring values. We then need to find two other grid values, here denoted as  $u_s$  and  $u_r$ , to form an upwinding stencil. Then  $u_o$  is found by minimizing a nonlinear expression derived from (1.9), using the values of  $u_r$ ,  $u_s$ , and  $f$ . The heap sort data structures are used in order to find  $u_r$  and  $u_s$ ; therefore, the complexity is  $N \log N$ , where  $N$  is the total number of grid points. Also, since  $u_r$  and  $u_s$  may not lie on the immediate neighbors, this algorithm may need a larger region around the initial wave front to get started.

As one will see in the following section, our proposed method is also based on following the characteristics. To update  $u_o$ , our method uses only the immediate neighboring grid values and does not need the heap sort data structure. More importantly, our algorithm follows the characteristics with certain directions simultaneously, in a parallel way, instead of a sequential way as in the fast marching method. The Godunov flux is essential in our algorithm, since it determines what neighboring grid values should be used to update  $u$  on a given grid node  $o$ . At least in the examples presented, we need only to solve a simple quadratic equation and run some simple tests to determine the value to be updated. This simple procedure is performed in each sweep, and the solution is obtained after a few sweeps. Our code is not much more than what is presented in section 3.2. We also point out the ease of implementing our proposed algorithm and its extension to more dimensions; this will be described in a sequel paper.

**2. A Godunov flux for strictly convex Hamiltonians.** By solving the Riemann problem for HJ equations (Godunov's procedure), Bardi and Osher [1] proved rigorously that

$$(2.1) \quad H_G(p_-, p_+; q_-, q_+) = \underset{p \in I[p_-, p_+]}{\text{ext}} \underset{q \in I[q_-, q_+]}{\text{ext}} H(p, q),$$

where

$$\begin{aligned} \underset{p \in I[a, b]}{\text{ext}} &= \min_{p \in [a, b]} \quad \text{if } a \leq b, \\ \underset{p \in I[a, b]}{\text{ext}} &= \max_{p \in [b, a]} \quad \text{if } a > b, \end{aligned}$$

$$H_G(D_-^x \phi_{ij}, D_+^x \phi_{ij}, D_-^y \phi_{ij}, D_+^y \phi_{ij}) = H_G(p_-, p_+; q_-, q_+),$$

and  $I[a, b]$  denotes the closed interval bounded by  $a$  and  $b$ . This is a monotone upwind flux function, which implies convergence. Godunov's scheme (1.3) for the eikonal equation  $\sqrt{\phi_x^2 + \phi_y^2} = 1$  can be derived from the above formula. It is one of the central topics of this paper to derive an explicit formula for the class of strictly convex Hamiltonians in question. Especially, we will demonstrate our numerical methods on  $H = \sqrt{a\phi_x^2 + b\phi_y^2 - 2c\phi_x\phi_y}$ ,  $c^2 < ab$ .

Note that, in general, if we reverse the order on  $p$  and  $q$  in our ext-ext decision, the result might be different, although they both give convergent monotone methods. However, in the convex Hamiltonian at hand, the results are order independent.

For convenience, we will also use  $H_G(\phi_{i,j}, \phi_{i\pm 1,j}, \phi_{i,j\pm 1})$  to denote the evaluation of our Godunov Hamiltonian  $H_G(D_-^x \phi_{ij}, D_+^x \phi_{ij}, D_-^y \phi_{ij}, D_+^y \phi_{ij})$ .

**2.1. Derivation of the flux.** In order to derive a compact expression that satisfies (2.1), we need to study the extremum of the Hamiltonian on  $I_p \times I_q \subset \mathbb{R}^2$ , where  $I_p$  is a shorthand for  $I[p_-, p_+]$ .

The extremum may occur on either the critical points of  $H$  or the boundary of  $I_p \times I_q$ . Let us first look at the partial derivatives of  $H$ , i.e.,  $H_p$  and  $H_q$ , and their zeros. Fix a  $q_0$ ; the extremum of  $H(p, q_0)$  occurs at either the critical point of  $H(p, q_0)$  (i.e., where  $H_p = 0$ ) or the boundary of  $I[p_-, p_+]$ . We denote the critical point by  $p_\sigma(q_0)$ . Similarly, given  $p_0$ , we obtain the critical point  $q_\sigma(p_0)$ . For convenience, we shall denote  $p_\sigma(q_0)$  by  $p_\sigma$  when  $q_0$  can be determined from the context, and  $(p_\sigma, q_\sigma)$  is the critical point of  $H$  such that  $H_p(p_\sigma, q_\sigma) = H_q(p_\sigma, q_\sigma) = 0$ . Therefore, we consider separately  $H(p_\sigma, q_\sigma)$ ,  $H(p_-, q_\sigma(p_-))$ ,  $H(p_+, q_\sigma(p_+))$ ,  $H(p_\sigma(q_-), q_-)$ ,  $H(p_\sigma(q_+), q_+)$ , and  $H(p_\pm, q_\pm)$  as possible evaluations of (2.1).

For fixed  $p$ , we have

$$(2.2) \quad H_G(p, q_-, q_+) = H(p, \operatorname{sgn} \max\{(q_- - q_\sigma)^+, (q_+ - q_\sigma)^-\} + q_\sigma),$$

where

$$\begin{aligned} \operatorname{sgn} \max(x, y) &= x^+ \quad \text{if } \max\{x^+, y^-\} = x^+, \\ \operatorname{sgn} \max(x, y) &= -y^- \quad \text{if } \max\{x^+, y^-\} = y^-. \end{aligned}$$

The expression for fixed  $q$  is a direct analogy to (2.2). It is easy to see that  $H_G(\cdot, \cdot; q_-, q_+)$  is increasing in  $q_-$  and decreasing in  $q_+$ . By symmetry,  $H_G(p_-, p_+; \cdot, \cdot)$  is increasing in  $p_-$  and decreasing in  $p_+$ .

Details of the derivation of the above expression are provided in the appendix.

The following proposition will be of use in analyzing this introduced Godunov flux.

**PROPOSITION 1.** *If  $H_{pp} > 0$ ,  $H_{qq} > 0$ ,  $pH_p \geq 0$ ,  $qH_q \geq 0$ , and  $p_\sigma(0) = q_\sigma(0) = 0$ , then  $p_\sigma(q) \equiv 0 \forall q$ , and  $q_\sigma(p) \equiv 0 \forall p$ .*

*Proof.*  $p_\sigma(q)$ , by definition, is the zero of  $H_p(p_\sigma(q), q) = 0$ . We will write  $p_\sigma$  in place of  $p_\sigma(q)$  for brevity. This proposition is then proved by simple manipulation of the definitions:

$$\begin{aligned} \frac{d}{dq} p_\sigma H(p_\sigma, q) &= p'_\sigma H_p(p_\sigma, q) + p_\sigma(H_{pp}(p_\sigma, q)p'_\sigma + H_{pq}(p_\sigma, q)) \\ &= p_\sigma p'_\sigma H_{pp}(p_\sigma, q) + \frac{\partial}{\partial q} H_p(p_\sigma, q) \\ &= p_\sigma p'_\sigma H_{pp}(p_\sigma, q) \\ &= 0. \end{aligned}$$

The hypothesis  $H_{pp} > 0$  implies that

$$p_\sigma(q) = 0 \quad \forall q \quad \text{or} \quad p'_\sigma(q) = 0 \quad \forall q.$$

Again, by the hypothesis that  $p_\sigma(0) = q_\sigma(0) = 0$ , we can conclude that  $p_\sigma(q) \equiv 0 \quad \forall q$ .

Similarly,  $q_\sigma(p) \equiv 0 \quad \forall p$ .  $\square$

Notice that if the Hamiltonian is  $\sqrt{p^2 + q^2}$ , our upwinding expression in (2.2) is identical to the conventional expression  $\max(p_-^+, p_+^-)$ . (In this case, the sign of the second argument does not matter since we are really evaluating its square product in the eikonal equation.) In fact, we have the following corollary, which is a direct consequence of Proposition 1.

**COROLLARY 1.** *If  $H_{pp} > 0, H_{qq} > 0, pH_p \geq 0, qH_q \geq 0, p_\sigma(0) = q_\sigma(0) = 0$ , and  $H(p, q) = H(|p|, |q|)$ , then the Godunov flux can be simplified to*

$$H_G(p_-, p_+; q_-, q_+) = H(\max\{p_-^+, p_+^-\}, \max\{q_-^+, q_+^-\}).$$

**3. The sweeping algorithms.** We will use the model equation (1.1) as a concrete example for the exposition of our algorithm. We stress here again that the scheme described below is valid for a general class of convex, homogeneous HJ equations.

From the assumption that the solution is nondecreasing along the characteristics, i.e.,

$$pH_p + qH_q \geq 0,$$

we can easily deduce that the solution is nondecreasing at least in either the  $x$ - or  $y$ -direction; i.e., either  $pH_p \geq 0$  or  $qH_q \geq 0$ . Since we approximate the derivatives  $\phi_x(x_{i,j})$  by finite differencing using the neighbors of  $\phi_{i,j}$ , the above monotonicity property translates to the following requirement in the solution  $\phi_{i,j}$ .

**DEFINITION 1.** *Let  $\phi_{i,j}$  be the solution of  $H_G(\phi, \phi_{i\pm 1,j}, \phi_{i,j\pm 1}) = r_{i,j}$ . We say that  $\phi$  satisfies the monotonicity requirement if*

$$\phi_{i,j} \geq \min\{\phi_{i\pm 1,j}, \phi_{i,j\pm 1}\}.$$

**3.1. Derivation of the scheme.** Without loss of generality, we assume that  $r(x, y) = 1$ . Let us reexamine the equation to be solved:

$$(3.1) \quad H(p, q) = 1,$$

where

$$H : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}.$$

Equation (3.1) dictates a level set relation; namely, the solution is the 1-level set of  $H$  in the  $p$ - $q$  plane (denoted here as  $\Lambda$ ). Correspondingly, the solutions of the HJ equation with the Godunov Hamiltonian

$$(3.2) \quad H_G(p_+, p_-; q_+, q_-) = \underset{p \in I[p_-, p_+]}{\text{ext}} \underset{q \in I[q_-, q_+]}{\text{ext}} H(p, q) = r$$

satisfy the following two properties:

- they are the intersections of  $\Lambda$  and the set  $I[p_-, p_+] \times I[q_-, q_+]$ ;
- they are either the critical points of  $H$  or the boundary points of the set  $I[p_-, p_+] \times I[q_-, q_+]$ .

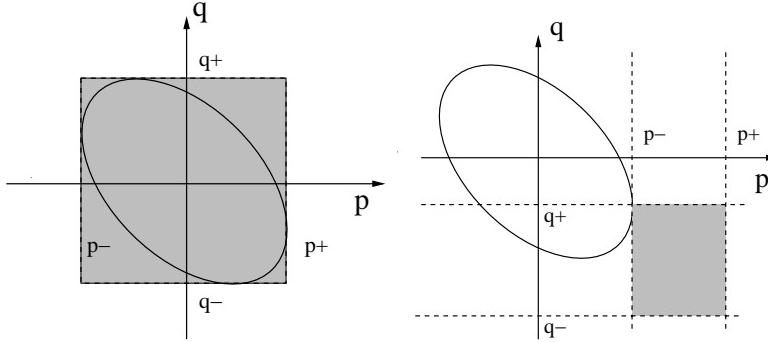


FIG. 1. The 1-level set of  $H$  and the box  $I[p-, p+] \times I[q-, q+]$ .

Figure 1 demonstrates two possible configurations of the intervals. So what our algorithm should do is find a suitable value of  $\phi$  on each grid node so that the divided forward and backward differences of  $\phi$  at that grid node satisfy (3.2).

Suppose we are on the grid node  $(i, j)$ , and it is determined that

$$H_G(p_+, p_-; q_+, q_-) = H(p_-, q_+) = 1.$$

Correspondingly, for our model equation (1.1) we have to solve the following quadratic equation:

$$(3.3) \quad a \left( \frac{\phi_{i,j} - \phi_{i-1,j}}{\Delta x} \right)^2 + b \left( \frac{\phi_{i,j+1} - \phi_{i,j}}{\Delta y} \right)^2 - 2c \left( \frac{\phi_{i,j} - \phi_{i-1,j}}{\Delta x} \right) \left( \frac{\phi_{i,j+1} - \phi_{i,j}}{\Delta y} \right) = 1.$$

The solution  $\phi_{i,j}$  not only has to satisfy the above equation, but ultimately has to be a solution to (3.2), given its four neighbors  $\phi_{i-1,j}$ ,  $\phi_{i+1,j}$ ,  $\phi_{i,j-1}$ , and  $\phi_{i,j+1}$ . The subfigure on the right in Figure 1 shows one such possible configuration; i.e.,

$$\frac{\phi_{i,j} - \phi_{i-1,j}}{\Delta x} < \frac{\phi_{i+1,j} - \phi_{i,j}}{\Delta x} \quad \text{and} \quad \frac{\phi_{i,j} - \phi_{i,j-1}}{\Delta y} < \frac{\phi_{i,j+1} - \phi_{i,j}}{\Delta y}$$

such that

$$\operatorname{ext}_{p \in I[p_-, p_+]} \operatorname{ext}_{q \in I[q_-, q_+]} H(p, q) = \min_{p \in I[p_-, p_+]} \min_{q \in I[q_-, q_+]} H(p, q) = 1.$$

One can, of course, implement a tree of all the probable cases from the complete listing of that of the Godunov Hamiltonian (2.1). However, we have a more straightforward approach that utilizes the compact expressions for the Godunov Hamiltonian (2.2) that we obtained from the previous section.

Instead, we solve the equation with the following reduced formulas for the original Godunov Hamiltonian:

$$(3.4) \quad H_G(p_+, p_-; q_+, q_-) = \operatorname{ext}_{q \in I[q_-, q_+]} H(p_-, q),$$

$$(3.5) \quad H_G(p_+, p_-; q_+, q_-) = \operatorname{ext}_{q \in I[q_-, q_+]} H(p_+, q),$$

$$(3.6) \quad H_G(p_+, p_-; q_+, q_-) = \underset{p \in I[p_-, p_+]}{\text{ext}} H(p, q_-),$$

$$(3.7) \quad H_G(p_+, p_-; q_+, q_-) = \underset{p \in I[p_-, p_+]}{\text{ext}} H(p, q_+),$$

$$(3.8) \quad H_G(p_+, p_-; q_+, q_-) = H(p_\sigma, q_\sigma).$$

For example, in the first case, the flux is equivalent to

$$H(p_-, \text{sgn} \max\{(q_- - q_\sigma)^+, (q_+ - q_\sigma)^-\} + q_\sigma) = 1.$$

The possible evaluations of  $\text{sgn} \max\{(q_- - q_\sigma)^+, (q_+ - q_\sigma)^-\} + q_\sigma$  are  $q_-$ ,  $q_+$ ,  $q_\sigma(p_-)$ , and 0. We thus end up solving the HJ equation with all possible arguments for the Hamiltonian.

Suppose we algebraically solve  $H(p_-, q_+) = 1$  for  $\phi_{i,j}$  and call the solution  $\phi^{can}$ . We then compute the divided differences  $p_\pm$  and  $q_\pm$  using this  $\phi^{can}$  in place of  $\phi_{i,j}$ . We call  $\phi^{can}$  valid if both

$$H(p_-, \text{sgn} \max\{(q_- - q_\sigma)^+, (q_+ - q_\sigma)^-\} + q_\sigma) = 1,$$

$$H(\text{sgn} \max\{(p_- - p_\sigma)^+, (p_+ - p_\sigma)^-\} + p_\sigma, q_+) = 1,$$

and  $\phi^{can}$  satisfies the monotonicity requirement (Definition 1).

Finally, we set  $\phi_{i,j}$  to be the minimum of those in the set of all valid candidate solutions  $\phi^{can}$  obtained from using all the possible combinations of the arguments of  $H$ . This is motivated by the first arrival time interpretation of the function  $\phi$ .

In essence, we are solving for the central value in the Godunov Hamiltonian in terms of its four neighbors. It is well known and easy to show that any monotone Hamiltonian, let alone Godunov's, is a monotone function of this value. For these Hamiltonians, this value goes from  $-\infty$  to  $+\infty$ . Thus there is always a unique solution.

**DEFINITION 2** (sweeping iteration). *A compact way of writing this sweeping iterations in C/C++ is the following:*

```
for(s1=-1;s1<=1;s1+=2)
for(s2=-1;s2<=1;s2+=2)
for(i=(s1<0?nx:0);(s1<0?i>=0:i<=nx);i+=s1)
for(j=(s2<0?ny:0);(s2<0?j>=0:j<=ny);j+=s2)
    update  $\phi_{i,j}.$ 
```

**3.2. The algorithm.** For the brevity of the algorithm, we define respectively

$$h_{G1}(p, q_-, q_+) := \text{sgn} \max\{(q_- - q_\sigma(p))^+, (q_+ - q_\sigma(p))^- \} + q_\sigma(p),$$

$$h_{G2}(p_-, p_+, q) := \text{sgn} \max\{(p_- - p_\sigma(q))^+, (p_+ - p_\sigma(q))^- \} + p_\sigma(q),$$

where  $q_\sigma(p) = pc/b$  and  $p_\sigma(q) = qc/a$ .

**ALGORITHM.** We assume that  $\phi(i, j)$  is given the exact values in a small neighborhood of  $\Gamma$ . We denote this neighborhood  $\text{Nbd}(\Gamma)$ . We initialize  $\phi$  by setting  $\phi(i, j) = \phi_{i,j}^{(0)}$  to  $\infty$ .<sup>1</sup> We begin by computing  $\phi_{i,j}^{(n)}$  for  $n = 1$ .

Do the following steps while  $\|\phi^{(n)} - \phi^{(n-1)}\| > \delta$  ( $\delta > 0$  is the given tolerance):

---

<sup>1</sup>Notice that we only need to use a large value in actual implementation.

1. For each grid point  $(i, j)$  visited in the sweeping iteration, if  $x_{i,j} \neq \text{Nbd}(\Gamma)$ , do the following:

- (a) For  $(s_x, s_y) = (-1, 1), (-1, -1), (1, -1)$ , and  $(1, 1)$

i. Solve

$$H\left(\frac{s_x \cdot (\phi_{tmp}(s_x, s_y) - \phi^{(n)}(i - s_x, j))}{dx}, \frac{s_y \cdot (\phi_{tmp}(s_x, s_y) - \phi^{(n)}(i, j - s_y))}{dy}\right) = r(i, j)$$

for  $\phi_{tmp}(s_x, s_y)$ .

ii. Let

$$p(s_x, s_y) = \frac{s_x \cdot (\phi_{tmp}(s_x, s_y) - \phi^{(n)}(i - s_x, j))}{dx}$$

and

$$q(s_x, s_y) = \frac{s_y \cdot (\phi_{tmp}(s_x, s_y) - \phi^{(n)}(i, j - s_y))}{dy}.$$

iii. Let  $T_{G1}(s_x, s_y)$  be the logical evaluation of the equality

$$H(p(s_x, s_y), h_{G1}(p(s_x, s_y), q(s_x, 1), q(s_x, -1))) = r(i, j),$$

and  $T_{G2}(s_x, s_y)$  be that of

$$H(h_{G2}(p(1, s_y), p(-1, s_y), q(s_x, s_y)), q(s_x, s_y)) = r(i, j).$$

iv. Let  $M(s_x, s_y) = \phi_{tmp}(s_x, s_y) - \min(\phi^{(n)}(i - s_x, j), \phi^{(n)}(i, j - s_y))$ .

v. If  $T_{G1}(s_x, s_y), T_{G2}(s_x, s_y)$  are true and  $M(s_x, s_y) \geq 0$ , add  $\phi_{tmp}(s_x, s_y)$  to the list `phi_candidate`.

- (b) For  $(s_x, s_y) = (1, 0), (-1, 0)$

i. Solve

$$H\left(\frac{s_x \cdot (\phi_{tmp}(s_x, 0) - \phi^{(n)}(i - s_x, j))}{dx}, \frac{s_x \cdot (\phi_{tmp}(s_x, 0) - \phi^{(n)}(i - s_x, j))}{dx} \frac{c}{b}\right) = r(i, j)$$

for  $\phi_{tmp}(s_x, s_y)$ .

ii. Compute  $p(s_x, s_y)$  and  $q(s_x, s_y)$ , following the definition.

iii. Evaluate  $T_{G1}(s_x, s_y)$ .

iv. If  $T_{G1}(s_x, s_y)$  is true and  $M(s_x, s_y) \geq 0$ , add  $\phi_{tmp}(s_x, s_y)$  to the list `phi_candidate`.

- (c) For  $(s_x, s_y) = (0, 1)$  and  $(0, -1)$

i. Solve

$$H\left(\frac{s_y \cdot (\phi_{tmp}(0, s_y) - \phi^{(n)}(i, j - s_y))}{dy}, \frac{s_y \cdot (\phi_{tmp}(0, s_y) - \phi^{(n)}(i, j - s_y))}{dy} \frac{c}{a}\right) = r(i, j)$$

for  $\phi_{tmp}(s_x, s_y)$ .

ii. Compute  $p(s_x, s_y)$  and  $q(s_x, s_y)$ , following the definition.

iii. Evaluate  $T_{G2}(s_x, s_y)$ .

iv. If  $T_{G2}(s_x, s_y)$  is true and  $M(s_x, s_y) \geq 0$ , add  $\phi_{tmp}(s_x, s_y)$  to the list `phi_candidate`.

- (d) Let  $\phi_{min}$  be the minimum element of `phi_candidate`.

$$\phi^{(n)}(i, j) = \min(\phi^{(n)}(i, j), \phi_{min}).$$

- (e) Clear phi\_candidate.  
 2. Set  $n = n + 1$ ; go back to step 1.

As described in the previous section, we have to solve the HJ equation with all possible arguments for the Hamiltonian and take the minimum of those in the set of all valid candidate solutions. The possible arguments of the Hamiltonian consist of the forward/backward differences of  $\phi$  and the critical points centered at each grid node. In the above algorithm, this set of all possible arguments is indexed by  $\{-1, 0, 1\}^2$ . Therefore, by  $X(-1, 1)$  we denote the quantity  $X$  that is computed using  $H(p_-, q_+)$ . The number 0 encodes the cases of critical points. For example,  $\phi_{tmp}(-1, 1)$  denotes the roots of the quadratic equation formed by  $H(p_-, q_+) = r$ ;  $\phi_{tmp}(1, 0)$  denotes that of  $H(p_+, q_\sigma(p_+))$ .

We remark that in the case of  $c = 0$ , our algorithm is equivalent to what is used in the fast marching method under the Rouy–Tourin formula (1.3). Secondly, in our numerical implementation, we put a threshold value in the evaluations  $T_{G1}$  and  $T_{G2}$  for numerical accuracy reasons.

**4. Examples.** Proposition 1 and Corollary 1 show the equivalence of the Godunov flux derived in this paper to the one commonly used in the fast marching applications. The use of this sweeping approach with the Godunov flux (1.3) has been reported in [24, 26] for eikonal equations; we will not repeat those examples in this paper. Instead, we present results of our algorithm applied to our model equation.

#### 4.1. Quadratic Hamiltonians $\sqrt{ap^2 + bq^2 - 2cpq}$ , $ab > c^2$ , $a, b > 0$ .

In each of the following examples, we compute the difference in the approximations in each successive iteration, i.e.,  $\|\phi^{n+1} - \phi^n\|_{L_1}$ , and say that the iterations have converged if this distance is less than  $\varepsilon \Delta x$ , where  $\varepsilon > 0$  and  $\Delta x$  is the grid size. In the examples presented in this paper, we simply set the threshold to be  $10^{-10}$ . Notice also that the set  $\Gamma$ , on which  $\phi = 0$ , is either a rectangle, an L-shaped piecewise linear object, or a set of isolated points. The reader can identify their location easily from the figures.

We started out by testing our algorithm on constant coefficient cases. In the case

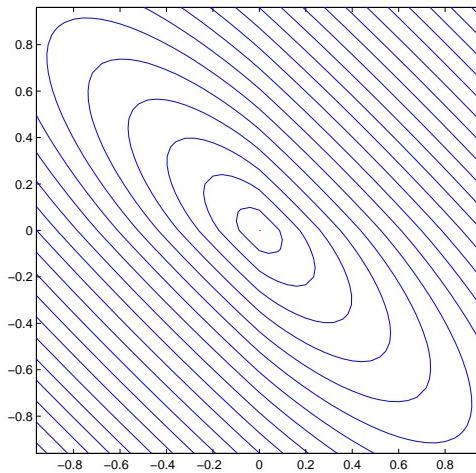


FIG. 2. A sweeping result after 2 sweeping iterations on a  $50 \times 50$  grid. The initial boundary is a single point in the center.  $a = 1.0$ ,  $b = 1.0$ ,  $c = 0.9$ .

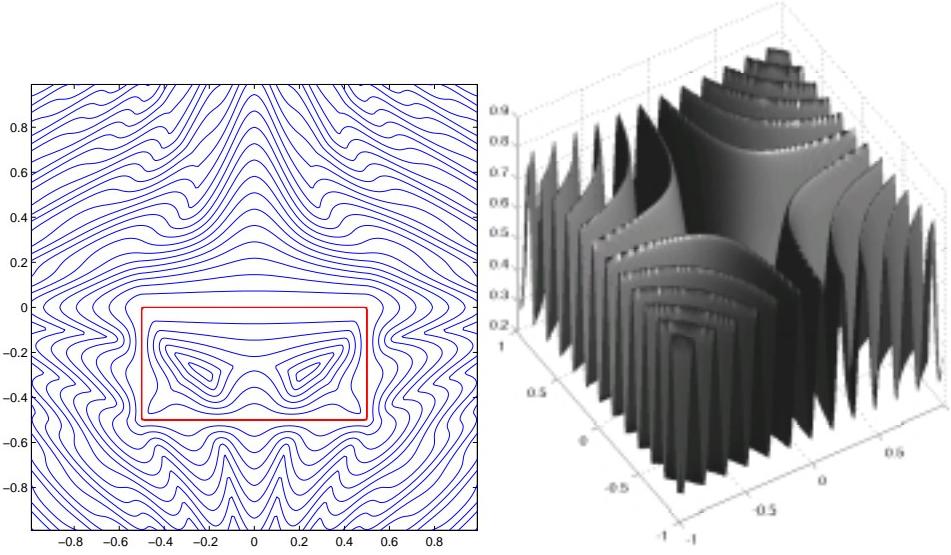


FIG. 3.  $a = 1, b = 1, c = 0$ , with a more oscillatory  $r(x) = 2.1 - \cos(4\pi^2 xy)$ , on a  $200 \times 200$  grid; convergence is reached in 7 sweeping iterations. The subplot on the left is the contour of the solution started with the square in the center. On the right is the graph of  $r(x)$ . Level curves with step 0.02 are plotted.

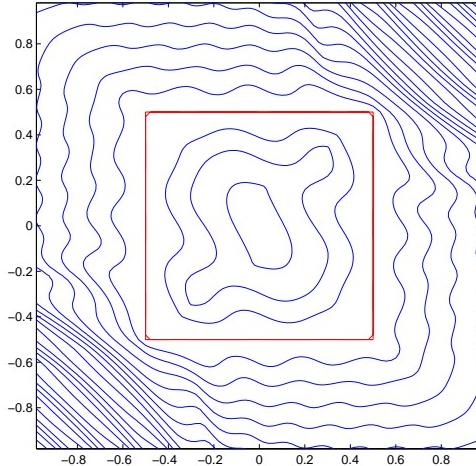


FIG. 4. (A very degenerate case)  $a = 0.375, b = 0.25, c = 0.29$ , with a more oscillatory  $r(x) = (2.1 - \cos(4\pi^2 xy))/4.0$ , on a  $100 \times 100$  grid. Notice that, in this case,  $ab = 0.0938$  is barely greater than  $c^2 = 0.0841$ . The contour of the solution is plotted. Convergence is reached at 43 sweeping iterations.

of  $a = b, c = 0$ , we have solutions that match the fast marching solutions. Figure 2 shows a result of a computation of the anisotropic case in which  $a = b = 1, c = 0.9$ . This is our first example in which the fast marching method is not applicable.

Next we apply the sweeping algorithm directly to cases in which the coefficients of the quadratic Hamiltonian or the right-hand sides are not constant. Figure 3 shows a computational result on a constant coefficient isotropic Hamiltonian and rather oscillatory forcing function. The rectangle in the middle is the set  $\Gamma$ . Figure 4

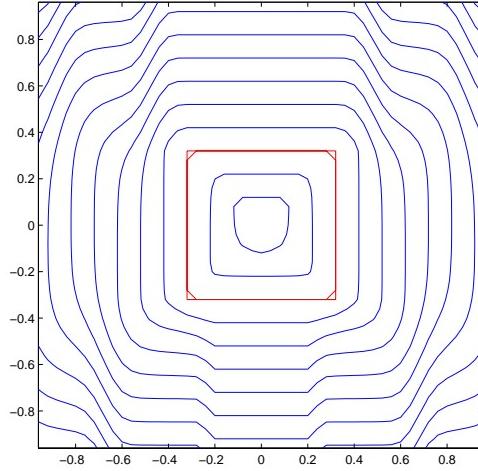


FIG. 5.  $a = 1$ ,  $b = 1$ ,  $c(x, y) = 0.9 \sin(5\pi x)$ , and  $r(x, y) = 1$ , on a  $50 \times 50$  grid. Convergence occurs after 10 iterations.

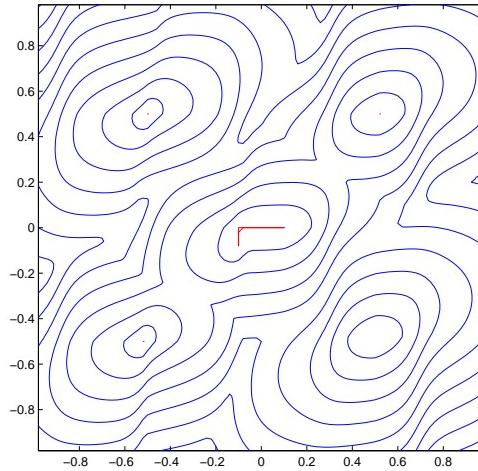


FIG. 6.  $a = 1.5 + \sin(5\pi x)$ ,  $b = 1$ ,  $c = -0.6$ , on a  $50 \times 50$  grid. Convergence occurs after 10 iterations.

shows a computational result for a very anisotropic case. We notice that the number of iterations needed for convergence seems to depend on the anisotropy of the Hamiltonian and also on how oscillatory the forcing term is. Figures 5, 6, and 7 show results obtained from variable coefficient Hamiltonians with constant and variable forcing function  $r(x, y)$ .

**4.2. Examples of distance on manifolds.** We now apply our sweeping algorithm to compute the geodesic distance on manifolds that are the graphs of certain functions. Given a function  $f(x, y)$ , with graph  $z = f(x, y)$ , we compute the coefficients  $a(x, y)$ ,  $b(x, y)$ , and  $c(x, y)$  according to (1.5) and apply our algorithm directly to the corresponding HJ equation. We first test the algorithm on a half-sphere with radius one. Figures 8 and 9 show the equidistance lines to one and two seed points,

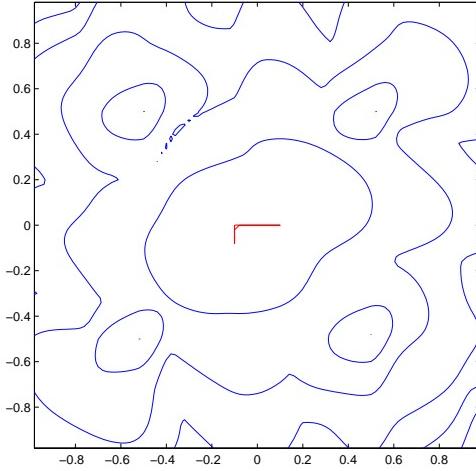


FIG. 7.  $a = 1.5 + \sin(5\pi x)$ ,  $b = 1$ ,  $c = -0.6$ , and  $r(x, y) = 2.1 + \cos(4\pi xy)$ , on a  $100 \times 100$  grid. Convergence occurs after 10 iterations.

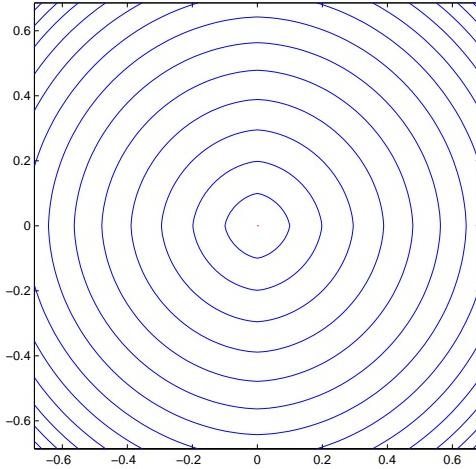


FIG. 8. This is an example of the distance on a half-sphere. The sweeping algorithm was applied to the graph of  $f(x, y) = \sqrt{1.0 - (x^2 + y^2)}$ , with  $\phi(0, 0) = 0$  as boundary condition, on a  $100 \times 100$  grid.

respectively. Figures 10, 11, and 12 show similar computation results applied to somewhat more oscillatory manifolds. As we expected, more sweeping iterations are required for convergence.

**4.3. Grid effects.** We first perform a rotation of the coordinate system. We represent this by

$$(x, y) \mapsto (\tilde{x}, \tilde{y})$$

and let

$$(a, b, c) \mapsto (\tilde{a}, \tilde{b}, \tilde{c}).$$

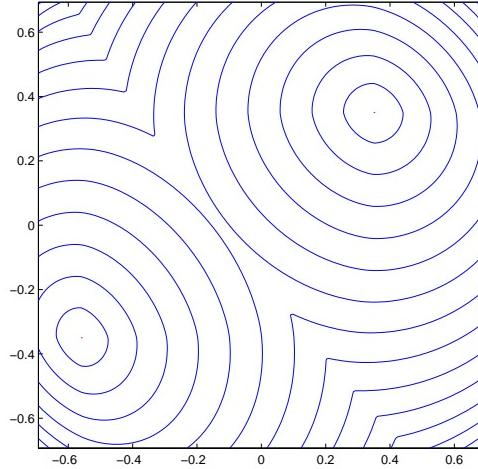


FIG. 9. This is an example of the distance on a half-sphere. The sweeping algorithm was applied to the graph of  $f(x, y) = \sqrt{1.0 - (x^2 + y^2)}$ , with two seed points. Convergence is reached after 2 sweeping iterations.

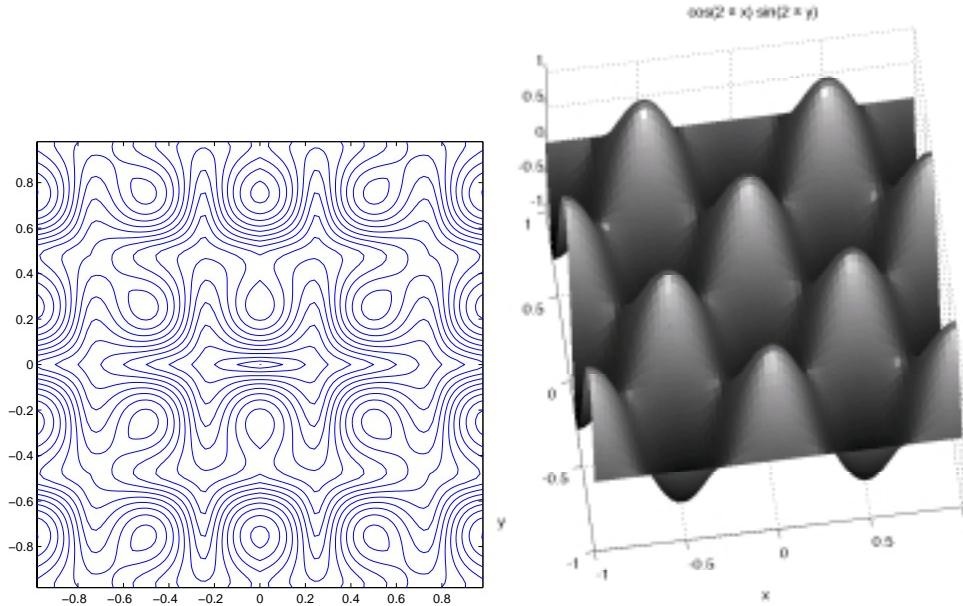


FIG. 10. The distance contour from the seed point  $(0,0)$  on the graph of  $f(x, y) = \cos(2\pi x) \sin(2\pi y)$ , on a  $100 \times 100$  grid. Convergence occurs after 9 iterations.

To study the grid effects of our sweeping algorithm, we set  $u = 0$  on a rotated square whose sides do not align with the grid lines. Comparing the results, shown in Figure 13, we see that the second picture, concentrating especially on the diamond-shaped contour in the middle, indeed shows grid effects compared to the first picture. However, with further grid refinement, as shown in the third picture, grid effects become unnoticeable, and the solution from our sweeping algorithm accurately ap-

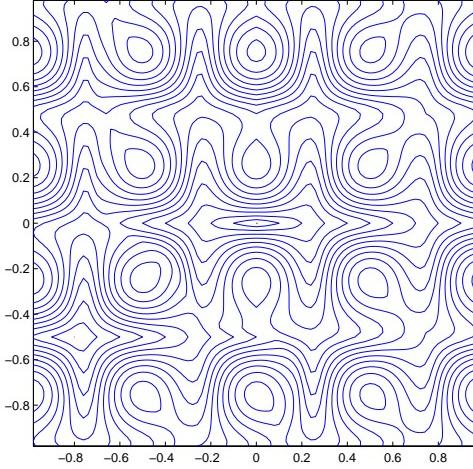


FIG. 11. The distance contour from the seed point  $(0, 0)$  and  $(-0.8, -0.5)$  on the graph of  $f(x, y) = \cos(2\pi x) \sin(2\pi y)$ , on a  $100 \times 100$  grid. Convergence occurs after 11 iterations.

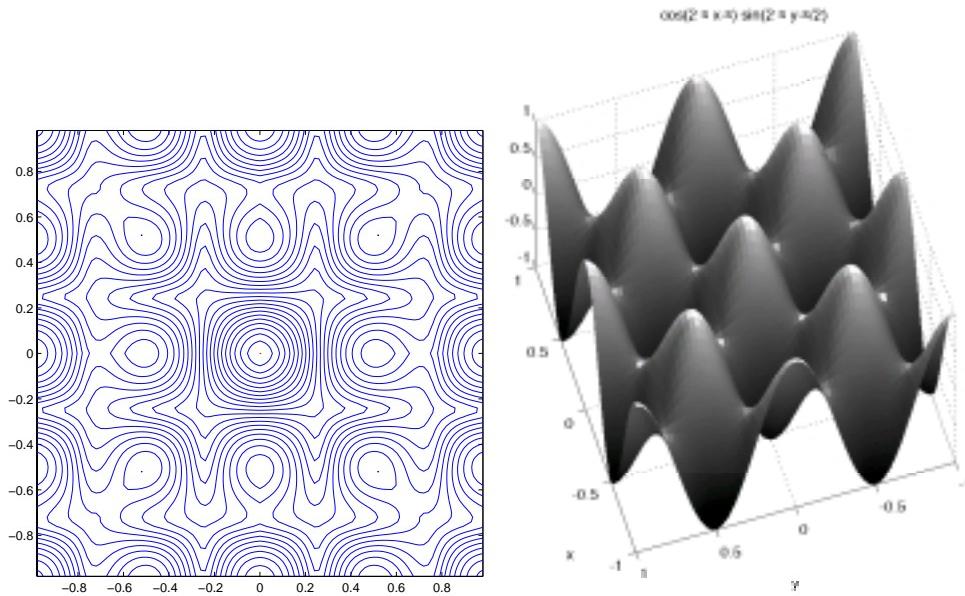


FIG. 12. The distance contour from the seed point  $(0, 0)$  on the graph of  $f(x, y) = \cos(2\pi x - \pi) \sin(2\pi y - \pi/2)$ , on a  $100 \times 100$  grid. Convergence occurs after 9 iterations.

proximates the exact solution.

**4.4. Comparison with the time marching solutions.** We use the first order Runge–Kutta–Lax–Friedrichs method [18] to discretize the following equation and march to steady state:

$$(4.1) \quad \tilde{\phi}_t + \operatorname{sgn}(\phi(x, y))(H(x, y, \tilde{\phi}_x, \tilde{\phi}_y) - r(x, y)) = 0,$$

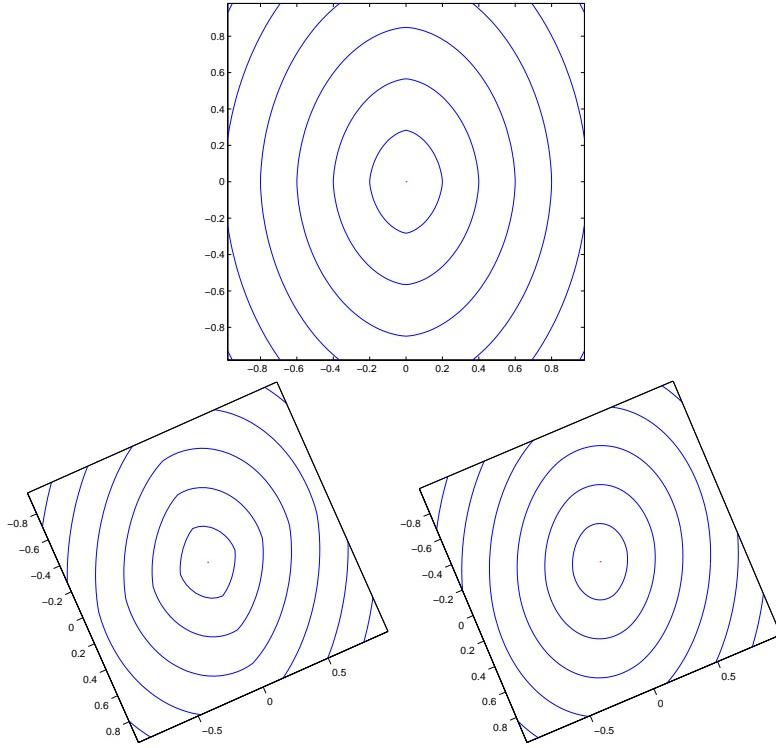


FIG. 13. Anisotropic case with a point source at  $(0,0)$ .  $a = 1, b = 1, c = 0.9$  and  $\tilde{a} = 1.70365$ ,  $\tilde{b} = 0.296352$ , and  $\tilde{c} = -0.561141$ , on  $50 \times 50$  and  $100 \times 100$  grids. Convergence occurs after 2 iterations.

TABLE 1  
Comparison of the time marching and sweeping solutions to the example shown in Figure 12.

	$dx = 2/50$	$2/100$	$2/200$	$2/400$	$2/800$
$\ \phi - \tilde{\phi}\ _{L_1}$	2.85423	1.83377	1.04008	0.56206	0.295738
$\ \phi - \tilde{\phi}\ _\infty$	1.03825	0.708986	0.436469	0.246439	0.133858

where  $\tilde{\phi}(x, y, t = 0) = \phi(x, y) = 0$  for  $(x, y) \in \Gamma$  and  $\phi$  is the solution obtained from the sweeping algorithm.

We remark that solving (4.1) is by no means a practical method for solving the steady state equation. Thousands of iterations are required for steady state, even if we take  $\phi$  as the initial Cauchy data. We use it only to verify the validity of our algorithm. Secondly, the solutions of (4.1) suffer from excessive smearing due to the numerical viscosity introduced by the Lax–Friedrichs method. As a consequence,  $\tilde{\phi}$  does not match well with  $\phi$  on coarse grids. The reader can compare Figure 14 with Figure 12, for example. However, we do see that  $\|\phi - \tilde{\phi}\|$  decreases with the refinement of the grid size; see Table 1 and Figure 14. We remark that higher order approximation schemes such as RK3-WENO5 will greatly reduce the numerical viscosity; the reader is referred to [18]. Our purpose here is only to show that the sweeping approximations converge to the viscosity solution.

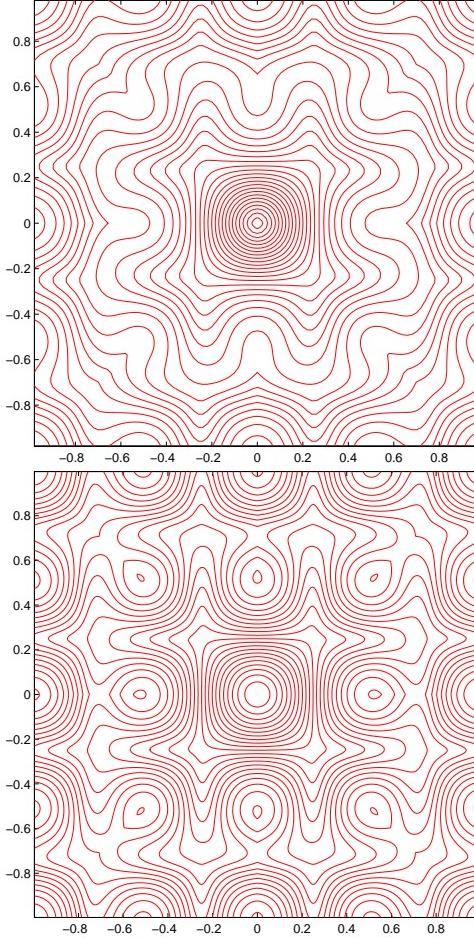
FIG. 14. Steady state of the time marching on a  $100 \times 100$  and  $800 \times 800$  grid.

TABLE 2

A numerical convergence study of the sweeping algorithm applied to the graph of  $f(x, y) = \sqrt{1.0 - (x^2 + y^2)}$ , with  $\phi(0, 0) = 0$  as boundary condition on the domain  $[-0.7, 0.7] \times [-0.7, 0.7]$ .

	$dx = 1.4/200$	$1.4/400$	$1.4/800$	$1.4/1600$
$  \phi - \tilde{\phi}  _{L_1}$	0.0138803	0.0079927	0.00453004	0.00253513
rate		0.796	0.819	0.84

**4.5. Numerical convergence.** Since we can easily compute the geodesic distance on a sphere, we will use it as an example to show numerical convergence of our algorithm. A distance contour plot is shown in Figure 8. Table 2 shows a numerical convergence of order 1. We have also noticed that the number of iterations needed for the  $L_1$  difference of the approximations in each successive iteration to decrease below the given tolerance seems to be bounded independently of the grid size. This number seems to depend on the anisotropy ( $c^2/ab$ ), the forcing function  $r$ , and the configuration of the interface  $\Gamma$ .

**5. Conclusion.** In this article, we studied a fast method for solving a class of time independent HJ equations with Dirichlet boundary conditions. The Hamiltonians of interest are homogeneous and convex. This fast method combines the idea of tracing the characteristics with Godunov construction and Gauss–Seidel iterations with smart choices of different updating sequences. In particular, we discussed some important properties of the Hamiltonian  $H = \sqrt{ap^2 + bq^2 - 2cpq}$ ,  $c^2 < ab$ , and the corresponding HJ equations. By the simple structure of the convexity, we derived a compact expression for the Godunov Hamiltonian that involves taking extrema of the Hamiltonian in relation to the evaluations of the derivatives of the solution. With our compact Godunov flux, the complexity of evaluating the Godunov Hamiltonian is reduced to only eight cases in two space dimensions. We then incorporated the expression into a simple Gauss–Seidel-type iteration procedure. We have produced some computational results using this algorithm. In particular, we have applied our algorithm to compute geodesic distances on graphs of functions. This is of some importance since people are interested in finding the geodesics on terrain-like manifolds.

We also remark that this Godunov-flux sweeping approach can be extended to higher dimensional cases. We are currently preparing another paper on this subject.

Our experience shows that the number of iterations needed depends on the amount of anisotropy and the nature of the forcing function. Under normal nondegenerate circumstances, experience shows an  $\mathcal{O}(N)$  complexity for convergence, where  $N$  is the number of grid points. Recently, in [26], the author provided some theoretical evidence on the bound of the number of iterations for isotropic, homogeneous eikonal equations. This points out a future research direction of bounding the number of sweeping iterations needed for convergence in relation to the anisotropy.

## 6. Appendix.

**6.1. Derivation of the flux for homogeneous convex Hamiltonians.** To obtain the formula used earlier in this paper, we simply verify its equivalence to the following cases, which rely only on the convexity of  $H$ :

$p_- < p_+$ , and  $q_- < q_+$ :

$$H_G = \min_{p \in [p_-, p_+]} \min_{q \in [q_-, q_+]} H(p, q).$$

- If  $q_\sigma \in [q_-, q_+]$ ,
  - $p_\sigma < p_- < p_+$ ,  $H(p_-, q_\sigma)$ ,
  - $p_- < p_+ < p_\sigma$ ,  $H(p_+, q_\sigma)$ ,
  - $p_- < p_\sigma < p_+$ ,  $H(p_\sigma, q_\sigma)$ .
- If  $q_\sigma < q_-$ ,
  - $p_\sigma < p_- < p_+$ ,  $H(p_-, q_-)$ ,
  - $p_- < p_+ < p_\sigma$ ,  $H(p_+, q_-)$ ,
  - $p_- < p_\sigma < p_+$ ,  $H(p_\sigma, q_-)$ .
- If  $q_\sigma > q_+$ ,
  - $p_\sigma < p_- < p_+$ ,  $H(p_-, q_+)$ ,
  - $p_- < p_+ < p_\sigma$ ,  $H(p_+, q_+)$ ,
  - $p_- < p_\sigma < p_+$ ,  $H(p_\sigma, q_+)$ .

$p_- < p_+$ , and  $q_- > q_+$ :

$$H_G = \min_{p \in [p_-, p_+]} \max_{q \in [q_+, q_-]} H(p, q) = \min_{p \in [p_-, p_+]} \max\{H(p, q_-), H(p, q_+)\}.$$

- If  $q_\sigma < q_+$ ,
  - $p_\sigma < p_- < p_+, H(p_-, q_-),$
  - $p_- < p_+ < p_\sigma, H(p_+, q_-),$
  - $p_- < p_\sigma < p_+, H(p_\sigma, q_-).$
- If  $q_\sigma > q_-$ ,
  - $p_\sigma < p_- < p_+, H(p_-, q_+),$
  - $p_- < p_+ < p_\sigma, H(p_+, q_+),$
  - $p_- < p_\sigma < p_+, H(p_\sigma, q_+).$
- If  $q_+ < q_\sigma < q_-$ ,
  - $(q_\sigma - q_+) > (q_- - q_\sigma), H(\cdot, q_+),$
  - $(q_\sigma - q_+) \leq (q_- - q_\sigma), H(\cdot, q_-).$

$p_- > p_+$ , and  $q_- > q_+$ :

$$H_G = \max_{p \in [p_+, p_-]} \max_{q \in [q_+, q_-]} H(p, q).$$

- If  $q_\sigma > q_-$ ,
  - $p_\sigma > p_-, H(p_+, q_+),$
  - $p_\sigma < p_+, H(p_-, q_+).$
- If  $q_\sigma < q_+$ ,
  - $p_\sigma > p_-, H(p_+, q_-),$
  - $p_\sigma < p_+, H(p_-, q_-).$
- If  $q_+ < q_\sigma < q_-$ 
  - $(q_\sigma - q_+) > (q_- - q_\sigma), H(\cdot, q_+),$
  - $(q_\sigma - q_+) \leq (q_- - q_\sigma), H(\cdot, q_-).$

$p_- > p_+$ , and  $q_- < q_+$ :

$$H_G = \max_{p \in [p_+, p_-]} \min_{q \in [q_-, q_+]} H(p, q).$$

- If  $q_\sigma \in [q_-, q_+]$ ,
  - $p_\sigma > p_-, H(p_+, q_\sigma),$
  - $p_\sigma < p_+, H(p_-, q_\sigma).$
- If  $q_\sigma < q_-$ ,
  - $p_\sigma > p_-, H(p_+, q_-),$
  - $p_\sigma < p_+, H(p_-, q_-).$
- If  $q_\sigma > q_+$ ,
  - $p_\sigma > p_-, H(p_+, q_+),$
  - $p_\sigma < p_+, H(p_-, q_+).$

**Acknowledgments.** YT thanks Paul Burchard and C.Y. Kao for their useful conversations on this topic. The authors also thank Prof. Giovanni Russo for his helpful suggestions to improve this paper.

#### REFERENCES

- [1] M. BARDI AND S. OSHER, *The nonconvex multi-dimensional Riemann problem for Hamilton-Jacobi equations*, SIAM J. Math. Anal., 22 (1991), pp. 344–351.
- [2] T. J. BARTH, *On the Marchability of Interior Stabilized Discontinuous Galerkin Approximations of the Eikonal and Related PDEs with Non-Divergence Structure*, NASA Technical report, NAS-01-010, NASA Ames Research Center, Moffett Field, CA, 2001.

- [3] M. BOUÉ AND P. DUPUIS, *Markov chain approximations for deterministic control problems with affine dynamics and quadratic cost in the control*, SIAM J. Numer. Anal., 36 (1999), pp. 667–695.
- [4] L.-T. CHENG, P. BURCHARD, B. MERRIMAN, AND S. OSHER, *Motion of curves constrained on surfaces using a level set approach*, J. Comput. Phys., 175 (2002), pp. 604–644.
- [5] M.G. CRANDALL AND P.L. LIONS, *Two approximations of solutions of Hamilton–Jacobi equations*, Math. Comp., 43 (1984), pp. 1–19.
- [6] M. G. CRANDALL AND P.-L. LIONS, *Viscosity solutions of Hamilton–Jacobi equations*, Trans. Amer. Math. Soc., 277 (1983), pp. 1–42.
- [7] P.-E. DANIELSSON, *Euclidean distance mapping*, Computer Graphics and Image Processing, 14 (1980), pp. 227–248.
- [8] J. HELMSEN, E. PUCKETT, P. COLELLA, AND M. DORR, *Two new methods for simulating photolithography development in 3d*, in SPIE 2726, Bellingham, WA, 1996, pp. 253–261.
- [9] G.-S. JIANG AND D. PENG, *Weighted ENO schemes for Hamilton–Jacobi equations*, SIAM J. Sci. Comput., 21 (2000), pp. 2126–2143.
- [10] J. B. KELLER, *Geometrical theory of diffraction*, J. Opt. Soc. Amer., 52 (1962), pp. 116–130.
- [11] R. KIMMEL AND J. A. SETHIAN, *Computing geodesic paths on manifolds*, Proc. Natl. Acad. Sci. USA, 95 (1998), pp. 8431–8435.
- [12] F. MEMOLI AND G. SAPIRO, *Fast computation of weighted distance functions and geodesics on implicit hyper-surfaces*, J. Comput. Phys., 173 (2001), pp. 730–764.
- [13] S. OSHER, *A level set formulation for the solution of the Dirichlet problem for Hamilton–Jacobi equations*, SIAM J. Math. Anal., 24 (1993), pp. 1145–1152.
- [14] S. OSHER AND R. P. FEDKIW, *Level set methods: An overview and some recent results*, J. Comput. Phys., 169 (2001), pp. 463–502.
- [15] S. OSHER AND J. HELMSEN, *A Generalized Fast Algorithm with Applications to Ion Etching*, manuscript.
- [16] S. OSHER AND B. MERRIMAN, *The Wulff shape as the asymptotic limit of a growing crystalline interface*, Asian J. Math., 1 (1997), pp. 560–571.
- [17] S. OSHER AND J. A. SETHIAN, *Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton–Jacobi formulations*, J. Comput. Phys., 79 (1988), pp. 12–49.
- [18] S. OSHER AND C.-W. SHU, *High-order essentially nonoscillatory schemes for Hamilton–Jacobi equations*, SIAM J. Numer. Anal., 28 (1991), pp. 907–922.
- [19] D. PENG, S. OSHER, B. MERRIMAN, AND H.-K. ZHAO, *The geometry of Wulff crystal shapes and its relations with Riemann problems*, in Nonlinear Partial Differential Equations (Proceedings of the Nonlinear PDE Emphasis Year meeting in Evanston, IL, 1998), G.-Q. Chen and E. DiBenedetto, eds., AMS, Providence, RI, 1999, pp. 251–303.
- [20] E. ROUY AND A. TOURIN, *A viscosity solutions approach to shape-from-shading*, SIAM J. Numer. Anal., 29 (1992), pp. 867–884.
- [21] J. A. SETHIAN AND A. VLADIMIRSKY, *Fast methods for the eikonal and related Hamilton–Jacobi equations on unstructured meshes*, Proc. Natl. Acad. Sci. USA, 97 (2000), pp. 5699–5703.
- [22] J. A. SETHIAN AND A. VLADIMIRSKY, *Ordered upwind methods for static Hamilton–Jacobi equations*, Proc. Natl. Acad. Sci. USA, 98 (2001), pp. 11069–11074.
- [23] J. A. SETHIAN, *Fast marching level set methods for three dimensional photolithography development*, in SPIE 2726, Bellingham, WA, 1996, pp. 261–272.
- [24] Y.-H. R. TSAI, *Rapid and accurate computation of the distance function using grids*, J. Comput. Phys., 178 (2002), pp. 175–195.
- [25] J. TSITSIKLIS, *Efficient algorithms for globally optimal trajectories*, IEEE Trans. Automat. Control, 40 (1995), pp. 1528–1538.
- [26] H.-K. ZHAO, *Fast sweeping method for eikonal equations I: Distance function*, SIAM J. Numer. Anal., submitted; also available at [www.math.uci.edu/~zhao](http://www.math.uci.edu/~zhao), 2002.
- [27] H.-K. ZHAO, S. OSHER, B. MERRIMAN, AND M. KANG, *Implicit and non-parametric shape reconstruction from unorganized points using variational level set method*, Computer Vision and Image Understanding, 80 (2000), pp. 295–319.